POSTER: I Don't Want That Content! On the Risks of Exploiting Bitcoin's Blockchain as a Content Store

Roman Matzutt, Oliver Hohlfeld, Martin Henze,
Robin Rawiel, Jan Henrik Ziegeldorf, Klaus Wehrle
Communication and Distributed Systems, RWTH Aachen University, Germany
lastname@comsys.rwth-aachen.de

ABSTRACT

Bitcoin has revolutionized digital currencies and its underlying blockchain has been successfully applied to other domains. To be verifiable by every participating peer, the blockchain maintains every transaction in a persistent, distributed, and tamper-proof log that every participant needs to replicate locally. While this constitutes the central innovation of blockchain technology and is thus a desired property, it can also be abused in ways that are harmful to the overall system. We show for Bitcoin that blockchains potentially provide multiple ways to store (malicious and illegal) content that, once stored, cannot be removed and is replicated by every participating user. We study the evolution of content storage in Bitcoin's blockchain, classify the stored content, and highlight implications of allowing the storage of arbitrary data in globally replicated blockchains.

1. INTRODUCTION

Bitcoin [9] and its underlying blockchain technology have revolutionized digital currencies and influenced other areas of research. Among the current cryptocurrencies, Bitcoin remains to be the most popular one with a market capitalization of $9.52 billion [6] and ≈5 200 active nodes [4] in August 2016. Motivated by its successful application in cryptocurrencies, the blockchain has been transferred to other domains and influenced a set of research areas. Examples include distributed naming services [1], digital notary services [7], and the realization of smart contracts [13]. This development highlights that blockchain technology constitutes both, a productively used paradigm securing sensitive assets and an active area of research.

While Bitcoin is designed as a cryptographic *currency*, the underlying blockchain technology can be (ab-)used in a more versatile manner—even as a general-purpose *content store*. Although a discussion of this possibility has been initiated by the community [8, 2] and few example contents have been identified [11], it still has received less attention than other, well-studied properties of Bitcoin (e.g., users' financial privacy [14, 15]) and a broad evaluation is still missing. Hence, we seek to fill this gap by providing a first step towards the *systematic* analysis of arbitrary content of Bitcoin's blockchain. Storing arbitrary content on a blockchain can be critical to the system's operability: users need to download the complete blockchain (or trust potentially malicious peers) to be able to participate. Further, as a blockchain constitutes a persistent and tamper-proof write-only log, its contents are virtually unerasable. Thus, the blockchain can be harmed by injecting arbitrary (and potentially malicious or illegal) content, which is not only unerasable, but also gets distributed among all Bitcoin users. As a consequence, users are put at risk by having to download illegal content; this harms the entire system. Our preliminary analysis reveals, e.g., that Bitcoin's blockchain contains roughly 400 links to illegal services, which each user stores on her hard disk.

We assess this risk by surveying different methods for data storage in Bitcoin's blockchain and by empirically analyzing the content in Bitcoin's blockchain. Our analysis is based on Bitcoin since it is the oldest and most widespread blockchain-based system and provides us with the largest dataset. We show the chronological development of content storage methods and discuss exemplary motivations of storing arbitrary data in the blockchain. This way we aim at opening a debate on the encouraged or discouraged use of the blockchain technology employed in a *single-purpose* system (e.g., Bitcoin) as *general-purpose* persistent content store.

2. STORING BLOCKCHAIN CONTENT

We briefly introduce the idea of blockchains and methods used to embed arbitrary content into them. Blockchains consist of a chain of cryptographically linked blocks holding use-case specific data. E.g., Bitcoin's blockchain is intended to hold monetary transactions from one user to another. Technically, Bitcoin uses a stack-based scripting language to specify the conditions under which bitcoins can be spent (in an *output script*). The funds can only be spent in new transactions containing an *input script* satisfying these conditions. Output scripts typically require a user to create a signature that can be verified with a specific public key. Although Bitcoin's scripting language is more powerful, the Bitcoin reference client only accepts *standard transactions* of a size of at most 100 KB that only use output scripts from a restricted set of templates. However, the peer appending a new block (the *miner*) can also decide to include *non-standard transactions* that do not follow the above-mentioned rules. Bitcoin allows for various ways to store arbitrary data on its blockchain, which we illustrate next.

CCS'16 October 24-28, 2016, Vienna, Austria

ACM ISBN 978-1-4503-4139-4/16/10.

DOI: http://dx.doi.org/10.1145/2976749.2989059

Coinbase transactions. Coinbase transactions reward miners with newly created bitcoins. As regular users cannot create such transactions, we consider them non-standard. The input script of a coinbase transaction encodes the block's position in the blockchain in a variable-length field. Inconsistencies between this length field and the length field of the input script can contain up to 100 Bytes [3] of arbitrary data. This is used by miners for, e.g., additional bytes of randomness [3] or for (unofficial[1]) feature voting.

P2PK(H) transactions. Pay-to-Pubkey (P2PK) and Pay-to-Pubkey-Hash (P2PKH) transactions are the most widely used standard transactions in Bitcoin. Only the private key corresponding to the public key (or hash value thereof) specified in the transaction's output script can spend the funds. Yet, users can exchange the public key (hash) with up to 65 Bytes (resp. 20 Bytes) of data per output script at the expense of *destroying* the bitcoins spent, i.e., making them permanently unspendable as the required private key is unlikely known by anybody. Considering a maximum size of 100 KB for standard transactions, a single transaction can hold up to 83.2 KB of arbitrary data distributed over 4 161 P2PKH outputs (98.3 KB using 1 513 obsolete P2PK outputs). *Multisig transactions*, standard transactions that are used to require *groups* of users to mutually agree on spending the funds, can be exploited in a similar vein.

Nulldata transactions. Added to the reference client as standard transactions in June 2013 (development version) and finally in March 2014 (release version), so-called *nulldata transactions* allow users to deliberately attach small pieces of data to regular transactions without destroying bitcoins. Nulldata transactions may only hold up to 83 bytes.

Non-standard transactions. Finally, output scripts can be extended with semantically irrelevant parts, e.g., dead if-branches or noneffective stack operations. Such transactions can combine content storage with a transaction behavior that is equivalent to, e.g., a P2PK(H) transaction, without destroying bitcoins. This method is the most space-efficient one as it allows for storing up to 99.6 KB for a transaction of standard size behaving like a P2PKH transaction. However, the majority of miners discard such transactions.

3. EVOLUTION OF CONTENT STORAGE

To assess the impact of arbitrary content on blockchain-based systems, we analyze the evolution of content stored on Bitcoin's blockchain. We employ heuristics for detecting transactions holding non-transactional data and extract *(i)* all *coinbase* transactions with ≥ 10 printable ASCII characters or known vote flags, *(ii)* all *P2PK(H)* and *multisig* transactions with at least 90% of printable ASCII characters (as suggested by CryptoGraffiti [10]), *(iii)* all *nulldata* transactions with non-empty payload, and *(iv)* all *non-standard transactions* containing output scripts not using a standard transaction template. Our coinbase and P2PK(H) transaction heuristics might occasionally lead to false positives with a probability of $< 10^{-5}$ for P2PKH transactions and coinbase transactions (which have a median length of 18 Byte).

Based on our heuristics, we analyzed the Bitcoin blockchain from its emergence in 2009 until end of July 2016 with a total of ≈146 million transactions (77.67 GB of data). We depict the rise of data storage per transaction type in Figure 1. In total, 0.80% of the transactions store data in the blockchain

[1]BIP 1 [12] defines the process for introducing new features

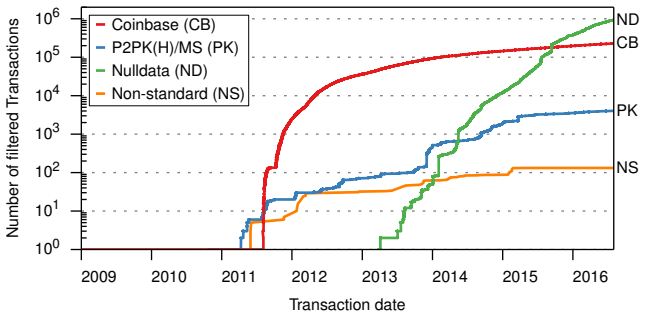

Figure 1: Evolution of number of transactions with arbitrary data for individual storage methods.

or deliberately deviate from standard transactions. Except for nulldata transactions (introduced in 2013), all content storage methods begin to spread in mid 2011, roughly six months after Bitcoin had started to be used effectively [5].

Besides a timestamped message engraved into Bitcoin's first block, first content was added using P2PKH transactions. The first such message, created on May 13, 2011, already shows the possibility to add files to the blockchain. The Bitcoin logo has been stored in two transactions [11], at the expense of 0.02 BTC, then worth $0.16 (now $10.30).

Our dataset of content-holding transactions contains 7 561 P2PK(H) and multisig transactions consisting of printable characters, holding text data of a total size of 1.86 MB. In future work, we plan to extend our analysis to also capture the more complex case of binary data being stored the blockchain, as done by services such as Apertus[2], as we expect such files to have a more impact on Bitcoin's blockchain.

As of version 0.9.0, released in June 2013, the reference client added nulldata transactions as a standard transaction type to store data on the blockchain. They were introduced to reduce the overhead of data-holding transactions as Bitcoin nodes actively keep track of unspent transactions. Since then, the usage of nulldata transactions experienced exponential growth, leading to a total of 936 174 nulldata transactions until end of July 2016. This growth is partly due to services[3] that link transactions to real-world assets via the Open Asset protocol. In total, 9.8% of all non-empty nulldata transactions can be attributed to a service using Open Assets.

Apart from the initial message, coinbase transactions started to contain data in the form of spiritual verses from August 5, 2011. The first advertisement was posted shortly after on August 25. Today, coinbase transactions often advertise the network (mining pool) that created the respective block. From January 2012 on, miners unofficially (cf. Section 2) voted on supporting Pay-to-Script transactions using coinbase transactions.

Surprisingly, we found a total of 288 716 non-standard transactions in the blockchain. However, the vast majority of 288 561 (99.9%) transactions consists of nulldata transactions with empty payload. Only 132 non-standard transactions do not use a standard script template. One of these transactions from August 2013 encodes a 1.45 KB patch for the reference client submitted by Satoshi Nakamoto in 2010.

We conclude that content is actively being stored on the blockchain. Notably, P2PK(H) transactions are still abused for storing content even though nulldata transactions serve

[2]http://apertus.io/

[3]e.g., https://www.coinprism.com/

as an officially supported, controlled way to store small amounts of data. We expect to obtain even more insights by attempting to detect binary files as well.

4. CLASSIFICATION OF CONTENT

So far, the Bitcoin community focused on identifying isolated examples of data stored on the blockchain [11]. Instead, we aim to *quantify* the properties of blockchain content and present preliminary results in the remainder of this paper.

We analyze those transactions mentioned in Section 3 that can be feasibly used to add larger amounts of data to the blockchain by arbitrary users in a timely manner, i.e., we exclude coinbase transactions from our analysis. In total, 940 354 (0.64% of the total) transactions matched this category. Of these, 99.6% are nulldata transactions of which 9.8% use the Open Assets protocol. A random manual inspection revealed attempted double spends, tweet-like short text messages, notary messages[4], and a large set of nulldata transactions that were not immediately attributable to a certain service. However, the manual inspection also revealed potentially critical and illegal content such as code segments, a leaked firmware private key, and the illegal prime number encoding a decryption algorithm for DVDs. In addition, we found 90 encrypted files and excerpts from the Tor Hidden Wiki containing links to indecent services (e.g., illegal pornography). Recall that *every* honest user unknowingly stores a copy of these contents, potentially exposing them to, e.g., prosecution.

Motivated by these insights, we searched our dataset for URLs as indicated by the keyword "http" as well as links to Tor hidden services as indicated by the ".onion" suffix. We observed 11 862 content-holding transactions containing URLs, which account for 1.3% of all transactions in the considered dataset. The majority of these transactions (11 659) used the nulldata, 193 the P2PK(H), 8 the multisig, and 2 non-standard storage methods. Here, messages from Open-Asset-based services account for ¾ of the links stored via nulldata transactions. The two non-standard transactions contain a JavaScript cross-site-scripting detector. Only the two transactions holding the Tor Hidden Wiki page dump contain links to content that must be considered illegal or at least highly questionable based on their descriptions. However, these two transactions contain almost 400 links. Such content may render downloading the blockchain illegal in certain jurisdictions [8], especially since images and PDF files have already been stored on the blockchain directly [11].

5. DISCUSSION

We showed that ≥0.80% of the transactions in Bitcoin's blockchain contain arbitrary content. While censorship-resistant content stores may be desirable in some cases, our analysis shows that this (ab)use puts Bitcoin at risk as a currency. This harm stems from the fact that *(i)* users must *locally replicate* the blockchain and *ii)* undesired content can never be removed. While most of the identified content appears to be included by honest services, the blockchain already contains arguably *problematic* content such as illegal code or links to illegal material. Moreover, illegal content could *deliberately* be injected to render using the blockchain (and thus using Bitcoin) illegal in certain jurisdictions [8], affecting *even honest users*. We therefore posit that a deeper understanding of the embedded content and methods preventing such content from being uploaded is necessary.

Towards this, future work involves gaining a deeper understanding of how content is injected and to broadly analyze blockchain content. This involves the content we already found as well as searching for additional content. Most importantly, we strive to address the more complex problem of broadly identifying and characterizing binary data, which is known from examples to exist in the blockchain. Furthermore, we plan to widen our analysis to other public blockchains and investigate the economical impacts of storing content on the blockchains of cryptographic currencies. Finally, we seek to develop countermeasures against the *uncontrolled* inclusion of data into these blockchains.

[4]e.g., https://docproof.org/ or https://bitproof.io/

Acknowledgments

This work has been funded by the German Federal Ministry of Education and Research (BMBF) under funding reference number 16KIS0443. The responsibility for the content of this publication lies with the authors.

6. REFERENCES

[1] M. Ali, J. Nelson, R. Shea, and M. J. Freedman. Blockstack: A Global Naming and Storage System Secured by Blockchains. In *USENIX ATC*, 2016.

[2] Bitcoin Wiki. Illegal content in the block chain. *https://en.bitcoin.it/wiki/Weaknesses*.

[3] Bitcoin Wiki. Transaction. *https://en.bitcoin.it/wiki/Transaction*.

[4] Bitnodes. Global Bitcoin Nodes Distribution. *https://bitnodes.21.co*.

[5] Blockchain.info. Bitcoin Charts. *https://blockchain.info/charts*.

[6] CoinMarketCap. Crypto-Currency Market Capitalizations. *https://coinmarketcap.com*.

[7] I. Eyal and E. G. Sirer. Majority is not enough: Bitcoin mining is vulnerable. In *FC*, 2014.

[8] E. McReynolds, A. Lerner, W. Scott, F. Roesner, and T. Kohno. Cryptographic Currencies from a Tech-Policy Perspective: Policy Issues and Technical Directions. In *Financial Cryptography and Data Security*. 2015.

[9] S. Nakamoto. Bitcoin: A peer-to-peer electronic cash system, 2008.

[10] Pseudonymous "Hyena". Bitcointalk Forum. *https://bitcointalk.org/index.php?topic=524877.msg9173767*.

[11] K. Shirriff. Hidden surprises in the Bitcoin blockchain and how they are stored: Nelson Mandela, Wikileaks, photos, and Python software. *http://www.righto.com/2014/02/ascii-bernanke-wikileaks-photographs.html*.

[12] A. Taaki. Bitcoin BIP 1. *https://github.com/bitcoin/bips/blob/master/bip-0001.mediawiki*, 2011.

[13] G. Wood. Ethereum: A Secure Decentralised Generalised Transaction Ledger. *Ethereum Project Yellow Paper*, 2016.

[14] J. H. Ziegeldorf, F. Grossmann, M. Henze, N. Inden, and K. Wehrle. Coinparty: Secure multi-party mixing of bitcoins. In *ACM CODASPY*, 2015.

[15] J. H. Ziegeldorf, R. Matzutt, M. Henze, F. Grossmann, and K. Wehrle. Secure and anonymous decentralized Bitcoin mixing. *Future Generation Computer Systems*, 2016.